



T.C.
SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
İŞLETİM SİSTEMLERİ PROJE RAPORU

B201210393- Enes KARAGÖZ

B191210552- Suhail Ali Mohammed Kamit

G191210553-Manar Saad Sadeq AlDoori

B191210574- Yaslam Bajandooh

G191210007- Özgenur Işık

GitHub: <https://github.com/SuhailTJ/Operating-System-Project>

SAKARYA

Aralık, 2023

İşletim Sistemleri Dersi

Özet

Java programlama dili kullanılarak sınırlı kullanılabilir kaynakların kısıtlamaları içinde çalışan dört seviyeli öncelikli proses görevlendiricisine sahip bir çoklu programlama sistemi simüle edilmiştir.

Not: Ödevde istenilen jar dosyası projenin **dist** klasörü içerisinde. **Github** <https://github.com/SuhailT J/Operating-System-Project>

Anahtar Kelimeler: Dispatcher, FCFS, proses, round robin, kuantum, geri besleme, java, zaman aşımı

GELİŞTİRİLEN YAZILIM

Program çalıştırıldığında giriş dosyasından okunan prosesler Dispatcher nesnesinde bulunun ve prosesleri tutan kuyruğa yerleştirilir. Proseslerin varış zamanı, öncelik değeri, işlenme zamanı gibi özellikleri vardır. Kuyruğa yerleştirilen prosesler özelliklerine göre sıralanarak uygun öncelikli kuyruklara yerleştirilir.

Gelen Procces'lerin Sıralanması ve Kuyruklara Alınması

Giriş dosyasından gelen process'leri öncelik değerlerine göre öncelikli kuyruklara yerleştiriyoruz. Dört öncelik değeri için dört adet kuyruğumuz bulunmakta.

Ardından process'ler karışık sırayla geldiği için ArrivalTime değişkenlerine göre sıraladık. Bu bizi process'leri her seferinde aramaktan kurtardı. Time değişkenine göre ilk elemana bakılıyor eğer ilk elemanın ArrivalTime'ı, Time değerimizden büyükse devamına bakmaya gerek kalmamaktadır. Sıralama fonksiyonunun içine bakacak olursak; Gelen kuyruğu SelectionShort algoritmasıyla beraber ArrivalTime'a göre sıraladık.

Görevlendiricinin Çalışması

Görevlendirici, sıfır öncelik değerine sahip prosesleri bulunduran kuyruğu gezmekle başlar. Görevlendiriciye gelen proses varsa bu proses FCFS algoritmasıyla işletilir. Sonlanana kadar devam eder. Bu kuyruk gezilirken aynı zamanda diğer kuyruklar da kontrol edilir. Kontrol için feedback() ve timeOutCheck fonksiyonları çağrılır. Eğer sıfır öncelikli proseslerin bulunduğu kuyrukta proses yoksa feedback fonksiyonunda önceki sırasına göre prosesler

çalıştırılır. Kuantum değeri bir olduğu için feedback’de her proses bir saniye çalışır ve prosesin önceliği düşürülerek askıya alınır. Askıya alınan prosesler farklı bir kuyrukta tutulur ve tekrar sıra kendine gelene kadar o kuyrukta bekler.

Tüm bu işlemler timer değişkeniyle senkronize edilir. `timeOutCheck` fonksiyonuyla da yirmi saniye çalışmayan prosesler tespit edilerek kuyruktan çıkarılır ve prosesin zaman aşımına uğradığını gösteren mesajlar konsola yazılır.

Olası İyileştirmeler

Görevlendiricide kullanılan sıralama algoritmasında daha efektif bir seçim yapabildik. Onun haricinde görevlendirici şuanda sadece 4 seviye için çalışmaktadır daha yüksek seviyelerde ek olarak kod yazmamız gerekli ve öncelik değiştirmek istediğimizde maliyeti yüksek olabilir.

Procces Sınıfı

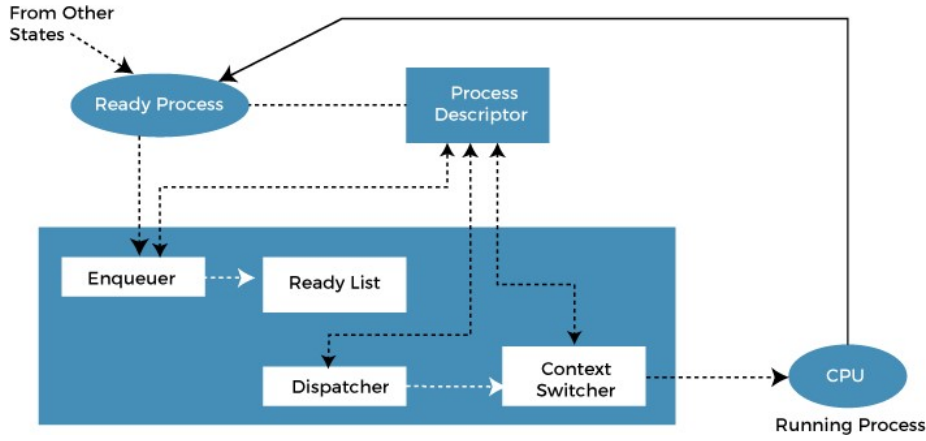
Dosyadan okunan her proses için kendi yazdığımız proses sınıfından nesneler oluştururuz ve gerekli değişkenlerin atamasını yaparız.

Görevlendirici tarafından prosesin çalışması için proses sınıfımızın `execute` fonksiyonu çağrılmalıdır. Bu fonksiyon Java proseslerini kullanarak daha önce oluşturduğumuz jar dosyasını çalıştırır ve konsola prosesin bilgileri yazdırılır. Java prosesi `Runtime.exec()` kullanılarak çağrılır.

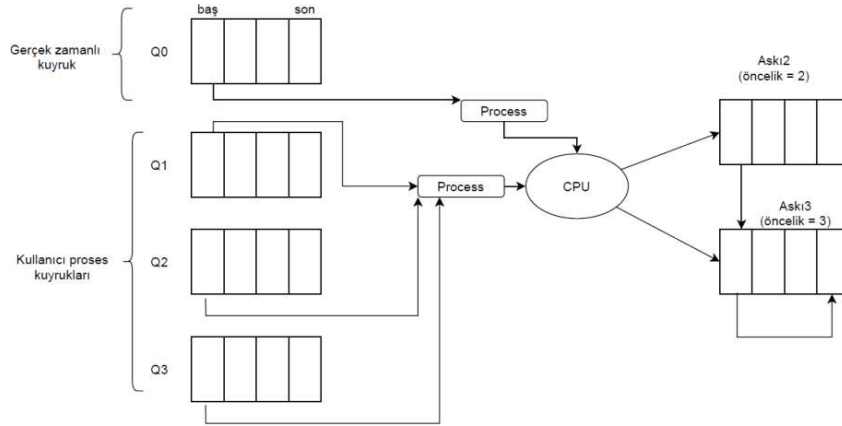
Proses mesajlarının renkli çıkabilmesi için `ColoredSystemOutPrintLn` sınıfındaki ANSI escape sequence’ları kullanılmıştır. Her yeni proses oluştuğunda `COLORS` array’indeki ANSI kodlarından biri o prosese özel bir `colorId` property’si şeklinde atanır. Bu `colorId`’ler, proses mesajlarının stringlerinde başa gelir. Proses mesajları ekrana renkli bir şekilde çıkar.

Görevlendirici Karşılaştırması

Görevlendirici modülünün işletim sistemi üzerindeki görevi gelen işlemlerin CPU'yu kullanımını düzenler, process'leri uygun pozisyonlara koyar ve kullanıcı modunu değiştirmeyi sağlar. Görevlendirici modülleri tek bir amaca hizmet etmediği için birden fazla çeşiti vardır. Bu yüzden kendi ihtiyaçlarımıza özel olarak görevlendirici algoritması yazabiliriz. İşletim sisteminin kullandığı görevlendirici şemasına bakalım olursak (bkz Görsel:1), gelen process'ler hazır kuyruğuna alınıyor ve sırası gelen process çalıştırılmak üzere CPU'ya yönlendiriliyor.



Bizim görevlendirici mekanizmamıza bakalım olursak :



Gelen processler önceliklerine göre cpu kullanımına izin veriliyor ve ardından önceliği 0 değilse bir alt seviyeye düşürülüp işleme devam ettiriliyor. Önceliği sıfır olmayan bir process işlem görürken sıfır öncelikli bir process gelirse işlem gören process askı kuyruğuna alınıyor ve sıfır öncelikli process'in işlemi bitene kadar bekliyor bekleme süresi 20 saniyeyi aşarsa process zaman aşımına uğrayıp siliniyor. Gerçek işletim sistemi process'leri alırken hazır kuyruğundan alıp işlettikten sonra aynı listenin sonuna tekrar ekleyerek devam etmekte biz bunu modellerken 2 adet askı kuyruğu kullandığımız için depolama performansımız düşük kaldı.

EKRAN ÇIKTISI

```
0 sn proses basladi (id:0 oncelik:1 kalan sure:2 sn Mbayt:574 yazicilar:2 tarayicilar:0 modemler:1 CDs:10)
1 sn proses askida (id:0 oncelik:2 kalan sure:1 sn Mbayt:574 yazicilar:2 tarayicilar:0 modemler:1 CDs:10)
1 sn proses basladi (id:1 oncelik:0 kalan sure:1 sn Mbayt:50 yazicilar:0 tarayicilar:0 modemler:0 CDs:1)
2 sn proses sonlandi (id:1 oncelik:0 kalan sure:0 sn Mbayt:50 yazicilar:0 tarayicilar:0 modemler:0 CDs:1)
(2 sn Gerçek-zamanlı? proses (64MB) tan daha fazla bellek talep ediyor - proses silindi)(2 sn Gerçek-zamanlı proses çok sayıda kaynak
talep ediyor ? proses silindi)(2 sn Gerçek-zamanlı proses çok sayıda kaynak talep ediyor ? proses silindi)(2 sn Gerçek-zamanlı pro
ses çok sayıda kaynak talep ediyor ? proses silindi)2 sn proses basladi (id:4 oncelik:2 kalan sure:2 sn Mbayt
:625 yazicilar:2 tarayicilar:1 modemler:1 CDs:1)
3 sn proses askida (id:4 oncelik:3 kalan sure:1 sn Mbayt:625 yazicilar:2 tarayicilar:1 modemler:1 CDs:1)
3 sn proses basladi (id:5 oncelik:2 kalan sure:3 sn Mbayt:765 yazicilar:0 tarayicilar:1 modemler:1 CDs:2)
4 sn proses askida (id:5 oncelik:3 kalan sure:2 sn Mbayt:765 yazicilar:0 tarayicilar:1 modemler:1 CDs:2)
4 sn proses basladi (id:9 oncelik:2 kalan sure:4 sn Mbayt:832 yazicilar:0 tarayicilar:1 modemler:0 CDs:1)
5 sn proses askida (id:9 oncelik:3 kalan sure:3 sn Mbayt:832 yazicilar:0 tarayicilar:1 modemler:0 CDs:1)
5 sn proses basladi (id:2 oncelik:3 kalan sure:2 sn Mbayt:515 yazicilar:0 tarayicilar:1 modemler:2 CDs:2)
6 sn proses askida (id:2 oncelik:3 kalan sure:1 sn Mbayt:515 yazicilar:0 tarayicilar:1 modemler:2 CDs:2)
(6 sn Proses (960 MB) tan daha fazla bellek talep ediyor ? proses silindi)6 sn proses basladi (id:11 oncelik:3 k
alan sure:2 sn Mbayt:853 yazicilar:2 tarayicilar:0 modemler:0 CDs:1)
7 sn proses askida (id:11 oncelik:3 kalan sure:1 sn Mbayt:853 yazicilar:2 tarayicilar:0 modemler:0 CDs:1)
7 sn proses basladi (id:12 oncelik:3 kalan sure:2 sn Mbayt:158 yazicilar:2 tarayicilar:0 modemler:1 CDs:2)
8 sn proses askida (id:12 oncelik:3 kalan sure:1 sn Mbayt:158 yazicilar:2 tarayicilar:0 modemler:1 CDs:2)
8 sn proses basladi (id:14 oncelik:1 kalan sure:4 sn Mbayt:78 yazicilar:1 tarayicilar:1 modemler:1 CDs:2)
9 sn proses askida (id:14 oncelik:2 kalan sure:3 sn Mbayt:78 yazicilar:1 tarayicilar:1 modemler:1 CDs:2)
(9 sn Proses (960 MB) tan daha fazla bellek talep ediyor ? proses silindi)(9 sn Proses (960 MB) tan daha fazla bellek talep ediyor
? proses silindi)(9 sn Proses (960 MB) tan daha fazla bellek talep ediyor ? proses silindi)9 sn proses basladi (
id:0 oncelik:2 kalan sure:1 sn Mbayt:574 yazicilar:2 tarayicilar:0 modemler:1 CDs:10)
10 sn proses sonlandi (id:0 oncelik:3 kalan sure:0 sn Mbayt:574 yazicilar:2 tarayicilar:0 modemler:1 CDs:10)
10 sn proses basladi (id:14 oncelik:2 kalan sure:3 sn Mbayt:78 yazicilar:1 tarayicilar:1 modemler:1 CDs:2)
11 sn proses askida (id:14 oncelik:3 kalan sure:2 sn Mbayt:78 yazicilar:1 tarayicilar:1 modemler:1 CDs:2)
(11 sn Proses (960 MB) tan daha fazla bellek talep ediyor ? proses silindi)(16 sn Proses (960 MB) tan daha fazla bellek talep ediy
o ? proses silindi)18 sn proses basladi (id:21 oncelik:3 kalan sure:2 sn Mbayt:379 yazicilar:0 tarayicilar:0 mo
demler:0 CDs:1)
19 sn proses askida (id:21 oncelik:3 kalan sure:1 sn Mbayt:379 yazicilar:0 tarayicilar:0 modemler:0 CDs:1)
```